Geo-Registration of Aerial Images using RANSAC Algorithm

Seema.B.S[1]
Department of E&C, BIT,
Bangalore, India,
seemabs810@gmail.com

Hemanth Kumar[2]
Department of E&C, BIT,
Bangalore, India,
drhkar2010@gmail.com

VPS Naidu[3]
MSDF Lab, FMCD,CSIR-NAL,
Bangalore, India,
vpnsnaidu@gmail.com

***Abstract*-The main objective of this paper is to determine the position of the target using geo registration technique with an onboard camera, in real time. The position estimation of the target is carried out with different algorithms or techniques; one of the algorithms is RANSAC feature detection. The benefits and the disadvantages of the algorithm are presented. The present approach of position estimation of the target is an alternative to commonly used INS/GPS approach. Vision based navigation uses a camera onboard to continuously transmit the aerial video to the ground control station (GCS) and the position of the target is determined using geo reference images such as those obtained from Google Earth.**

Keywords- RANSAC algorithm, Geo-registration, target position estimation.

I.INTRODUCTION

The RANdom SAmples Consensus (RANSAC) algorithm was proposed by Fischler and Bolles [1]. It is a general parameter estimation approach designed to cope with a large proportion of outliers in the input data. Unlike many of the common robust estimation techniques such as M-estimators and least-median squares that have been adopted by the computer vision community from statistic literature, RANSAC was developed from within the computer vision community [1].

The RANSAC (Random Sample Consensus) algorithm is a simple, yet a powerful, technique that is commonly applied to the task of estimating the parameters of a model, using data that may be contaminated by outliers. RANSAC estimates a global relation that fits the data, while simultaneously classifying the data into inliers (points consistent with the relation) and outliers (points not consistent with the relation)[2]. Due to its ability to tolerate a large fraction of outliers, the algorithm is popular choice for a variety of robust estimation problems. RANSAC operates in hypothesize and verify framework, a minimal subset of input data points is randomly selected and model parameters are estimated from this subset [3]. The model is then evaluated on the entire dataset and the number of data points consistent with the model is determined. RANSAC is used to find the correct solution even for high levels of contamination (noise), however, the number of samples required to do so increases exponentially, and the associated computational cost is substantial[4].

There have been a number of recent efforts aimed at increasing the efficiency of basic RANSAC algorithm where as only few efforts have been directed towards the goal of formulating RANSAC in manner that is suitable for real time implementation. A basic RANSAC algorithm uses the smallest set of data possible and proceeds to enlarge this set with consistent data points [5].

II. RANSAC ALGORITHM FOR GEO-REGISTRATION

The RANdom SAmple Consensus (RANSAC) algorithm is a feature based method of image matching that is more reliable than the edge based methods as they compute features which are rotation invariant and robust[6]. These features are classified into two i.e. inliers (points that match) and outliers (points that do not match). The RANdom SAmple Consensus (RANSAC) algorithm is used to find the features of the image and eliminate those that do not match (outliers)[7]. Features have to be robust, rotation invariant, invariant to distortions and obstructions and independent. The features can be selected manually for comparison but the UAV that has a camera on board cannot hover over a particular area. It has to be in constant motion. Therefore, the extraction and comparison will be automated and continuous. RANSAC algorithm gives accurate results even though there is scaling and rotation between the geo-referenced image and camera image at the cost of execution time[8].

III.METHODOLOGY

The planned activity for target position estimation consist of an onboard camera and transmitter which sends the video data and GPS which sends the navigational data i.e., latitude and longitude values to the ground station. In real, the camera must continuously transmit the video footage of the ground to the GCS where it is received and processed.

In Ground control, the latitude and longitude values that are received from the GPS are used to construct the URL query and generate the geo-reference image with latitude and longitude grid information.The geo-reference image covers the entire trajectory of the flight. The geo-reference image and the image received from the UAV will be processed and the position of the recovered UAV image will be determined using geometric transformation matrix using RANSAC algorithm. Once the position of the target is estimated from the geo-referenced images' latitude longitude grid or by spatially referencing the geo-referenced image the latitude and longitude values of the target is displayed on the click of the mouse pointer.

If the geo-referenced database can be constructed or purchased, then the image obtained by the onboard camera can be matched with the images in the database though this may be time consuming. If the initial coordinates and approximate flight trajectory are known, the process can be speeded up. However, the erroneous GPS coordinates are used to obtain a geo-referenced image from Google Static Maps where the area depends on the zoom level and the input coordinates are at the center.

IV. GEOMETRIC TRANSFORMATION

The Geometric Transform Estimator, estimates geometric transformation from matching point and returns the transform in a form of matrix. This transformation can be used to compute projective, affine, or non reflective similarity transformations with robust statistical methods, such as, RANSAC and Least Median of Squares. Non-reflective similarity transformation is used in this study.Non-reflective similarity transformations may include a rotation angle (θ=10 used for the study), a scaling, and a translation. Shapes and angles are preserved.It has four degrees of freedom and requires two pairs of points.

The transformation matrix is: $H = \begin{bmatrix} h_1 & -h_2 \\ h_2 & h_1 \\ h_3 & h_4 \end{bmatrix}$

The projection of a point $[x \quad y]$ by H is $[\hat{x} \quad \hat{y}] = [x \quad y \quad 1]H$

Where $h_1 = scalingfactor \times \cos(\theta)$

$h_2 = scalingfactor \times \sin(\theta)$

h_3= x-translation

h_4= y-translation

A. ALGORITHM

Step 1: The latitude and longitude values are received from the GPS onboard. The latitudes on the southern hemisphere and longitudes on the western hemisphere must be passed as negative values.

Step 2: Provide the same to Google Static Maps and construct the URL query and hence generate the geo-reference image which is one of the input for thisalgorithm another input is the UAV image which is captured from the onboard camera.

Step 3: The blob features (similarity features) are computed using detect feature function (detectSURFFeatures) which uses Speeded-Up Robust Feature(SURF) algorithm and then the features of both the image are extracted using Extract feature function (extractFeatures).

Step 4: Once the features are computed, they are matched using the Computer Vision Toolbox and matches are displayed.

Step 5: A variable 'count' is maintained to store the number of inliers after comparison with the geo-reference image. The minimum number of inliers to be found is 15. Inliers are consistent with the projection of the image [2].

Step 6: The position of the recovered UAV image is found using a geometric transform and the recovered UAV image is overlaid on the geo-referenced image.

Step 7: The geo-referenced image contains the latitude and longitude grid and thus the position can be estimated in coordinates.

Step 8: Once the position is estimated, the latitude and longitude values are displayed by the mouse click on the recovered UAV image.

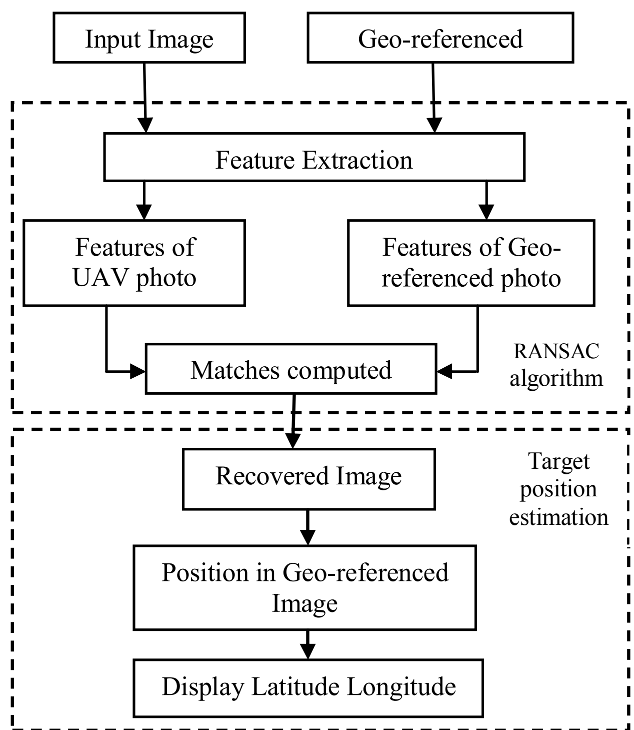


Fig 1.RANSAC Algorithm for Geo-registration and target position estimation

V. POSITION ESTIMATION

The UAV image is overlaid on the geo-reference image and the position of the UAV image matrix is stored in indices. In this section, the exact position estimation from the position of the UAV image in the geo-referenced one is discussed. The geo-referenced image contains information such as latitude and longitude pair at the center. First, the bounds of the geo-referenced image must be calculated. The pixel values at the center of the geo-referenced image can be calculated by converting the latitude and longitude values into meters and then from meters into pixels [9]. Latitude and longitude obtained from the geo-reference image will be in decimal degrees i.e., WGS84 (World Geodetic System) which are converted into meters using the following formulae,

$$x = \frac{(lon)(OS)}{180} \tag{1}$$

$$y = \frac{(y')(OS)}{180} \tag{2}$$

Where $y' = \dfrac{\log\left(\dfrac{\tan(90+lat)\pi}{360}\right)}{\left(\dfrac{\pi}{180}\right)}$

$$OS = \frac{R_e(2\pi)}{2}$$

Where R_e = Radius of the earth (6378137m).

OS = Original shift.

For a conversion from meters to pixels, the following conversion is used:

$$x_1 = \frac{(x+OS)}{res} \tag{3}$$

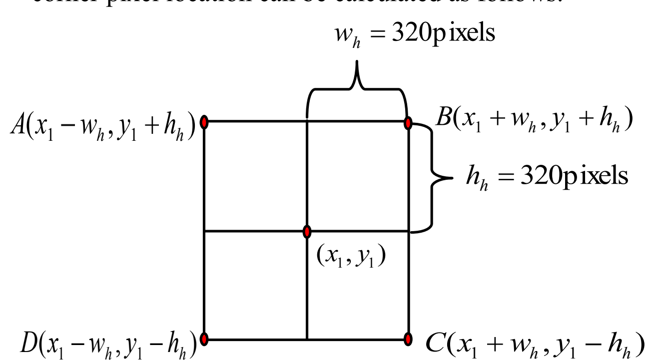
$$y_1 = \frac{(y+OS)}{res} \tag{4}$$

Where Resolution $= res = \dfrac{2\pi R_e}{2^{ZL}\,TS}$

TS = Tile size (320 X 320 used here).

ZL = Zoom level (17-19 for Google maps)

Thus, the pixel values at the center are determined. The corner pixel location can be calculated as follows:



Where width = height = 640 pixels

w_h = width/2 = 320 pixels

h_h =height/2 = 320 pixels

Longitude is represented in the x direction and latitude in represented in the y direction. Then, the corner pixels must be converted into meters and then converted to latitude and longitude using the following formulae:

$$A_{WGS}(lon,lat) = \left(\frac{(x_1 - w_h)res - OS}{OS}180, \frac{(y_1 + h_h)res - OS}{OS}180\right) \tag{5a}$$

$$B_{WGS}(lon,lat) = \left(\frac{(x_1 + w_h)res - OS}{OS}180, \frac{(y_1 + h_h)res - OS}{OS}180\right) \tag{5b}$$

$$C_{WGS}(lon,lat) = \left(\frac{(x_1 + w_h)res - OS}{OS}180, \frac{(y_1 - h_h)res - OS}{OS}180\right) \tag{5c}$$

$$D_{WGS}(lon,lat) = \left(\frac{(x_1 - w_h)res - OS}{OS}180, \frac{(y_1 - h_h)res - OS}{OS}180\right) \tag{5d}$$

Thus, the bounding values of the latitude longitude for the image are calculated. A spatial referencing object is created and the bounding values calculated in the previous step are assigned to it [10]. The size of the image (640x640) is also assigned. The position of the UAV image within the geo-referenced image is calculated. In feature detection method, the position of the recovered image after applying the geometric transform is stored[11,12]. Using the spatial referencing object and the known pixel values of the position as determined, the latitude longitude values can be computed using the pix2latlon () function in MATLAB. Once the position of the target is determined then the location is displayed on the click of the mouse pointer.

VI. RESULTS AND DISCUSSION

Algorithm uses geo-reference images taken from Google Earth and the UAV shot which is obtained from the onboard camera .The latitude and longitude pair used for all the testing purposes is 12.9653471, 77.651822703 respectively. The camera model is applied to obtain the UAV image (Fig. 2a). The correct inliers detected by RANSAC are compared (Fig. 2b). The UAV image is recovered by applying the detected transform (Fig. 2c). Thus, the position of the target in the geo-referenced one is known by overlaying the two (Fig. 2d). Once the position of the target is determined, the position of the target is displayed by a mouse click on the recovered UAV image.

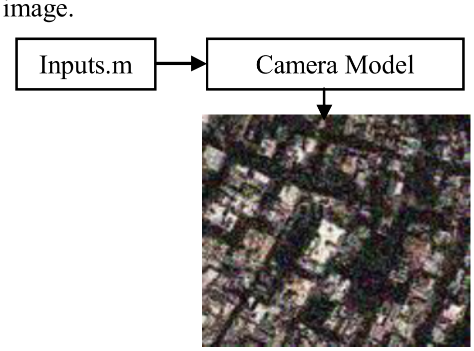


Fig. 2a. UAV image obtained from camera model

The algorithm functions well for up to 50° clockwise or anticlockwise, beyond which the error is too big as the transform recovered loses accuracy. The optimum rotation is observed to be 30°. The number of inliers falls with increase in noise.However, when there is no rotation, scaling or blur, the RANSAC algorithm gives an error of 0 hence the algorithm is accurate.

On performing Monte Carlo simulation it is found that the RANSAC algorithm fails in the event of variance of Gaussian noise being equal to or greater than 0.04 and the error in the latitude and longitude with respect to variance in noise is as shown in Table.1 and the number of inliers are found to drop drastically with increase in noise. The minimum requirement of inliers (15) is not met, and then the algorithm will no longer be able to recover the position of the image correctly and will fail.In Table-2 the functionality of RANSAC under different scaling factorsare shown. RANSAC fails for a scale factor of 6 or greater.

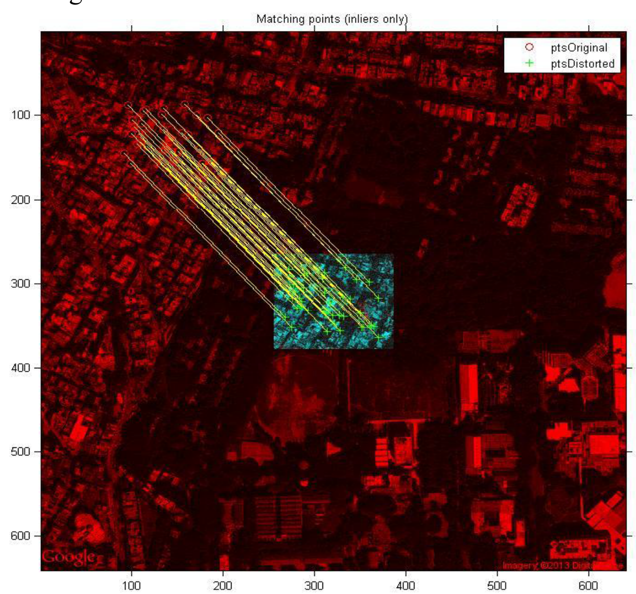


Fig. 2b.Inliers detected by RANSAC algorithm.


Fig.2c. Recovered Image.


Fig.2d. Position of target.

Table.1.Functionality of RANSAC under noise variance.

Noise	Inliers	Error latitude	Error longitude
0.00	39	-7.976e-06±7.39e-07	8.046e-06±0
0.01	33	-7.946e-06±7.39e-07	8.046e-06±0
0.02	29	-7.632e-06±7.39e-07	7.93e-06±1.47e-06
0.03	25	-7.632e-06±2.37e-06	8.69e-06±2.35e-06
0.04	13	-	-

Table.2. Functionality of RANSAC under scaling factors.

Scale	Inliers	Error latitude	Error longitude
1	42	-7.841e-06±7.39e-06	8.046e-06±0
2	38	-4.809e-06±2.60e-06	5.47e-06±2.70e-06
3	32	-3.032e-06±1.43e-06	3.04e-06±1.28e-06
4	25	-2.718e-06±1.39e-06	2.89e-06±1.06e-06
5	19	-2.613e-06±1.09e-07	2.682e-06±0
6	12	-	-

In Table. 3 the functionality of RANSAC under different theta value are shown. RANSAC provides accurate results for all the values of theta.The error tends to decrease as the theta value increases.

Table.3. Functionality of RANSAC under theta value.

theta	Inliers	Error latitude	Error longitude
0	59	-7.8414e-06±0	2.682e-06±0
10	38	-7.736e-06±7.39e-07	8.04e-06±2.61e-06
20	30	-7.094e-06±6.97e-06	7.73e-06±2.60e-06
30	25	-6.795e-06±2.24e-06	8.0466e-06±0
90	53	-2.6138e-06±0	2.6829e-06±0
180	59	-2.6138e-06±0	2.6829e-06±0

The functionality of RANSAC under different types of noise and the respective inliers detected and the error in latitude and longitude are given in Table.4.

Table.4. Functionality of RANSAC under different types of noise.

Noise	Inliers	Error latitude	Error longitude
Gaussian	39.84± 3.79	-7.84e-06±1.56e-06	8.046e-06±0
Poisson	37.96± 2.77	-7.84e-06±1.12e-06	8.046e-06±0
Salt & Pepper	29.82± 2.62	-7.84e-06±0	8.046e-06±0
Speckle	31.10± 3.32	-7.84e-06±0	8.046e-06±0

VII. CONCLUSION

The RANSAC algorithm for the target position estimation have been implemented and compared according to various parameters. Under ideal circumstances, the algorithm shows an error in the position to be 0. RANSAC (8.456 sec) takes twice the amount of time taken by Normalized Cross Correlation (NCC) (4.267sec) [13].It is observed that RANSAC produce accurate results when there is scaling and rotation between the geo-reference image and camera image at the cost of execution time,even though NCC is faster, it fails in the event of noise, scaling, rotation or blur [13].RANSAC can estimate the features with a high degree of accuracy even when a significant number of outliers are present. RANSAC produces an accurate overlay and can also be used in non ideal conditions. RANSAC is equipped for practical use and only fails in the event of extreme distortion or noise but is relatively invariant to rotation; noise and scaling as feature detectionmethods are more efficient than edge detection and brute force comparison methods.

RANSAC has no upper bound on the time it takes to compute the features. One of the main drawbacks of this algorithm is if minimum numbers of features are not detected then the algorithm has to be executed again. When the number of iterations computed is limited the solution obtained is not optimal. Different numbers of features are computed each time the algorithm is executed. RANSAC is preferred in practical application as it is a feature detection methods and hence produces accurate result when compared to edge detection and also robust in feature.

ACKNOWLEDGMENT

We would like to thank Mr. N Santhakumar, FMCD Department, CSIR-National Aerospace Laboratoryand Dr.K.V.Prasad, Head of the Department, Electronics& Communication Engineering, Bangalore Institute of Technology for their constant encouragement and support to carry out this work.

REFERENCE

[1] Markus Olgemar , ***"Camera based navigation: Matching between sensor reference and video image",*** Master Thesis, Linkopings University, 2008.

[2] Daniele Fontanelli and Luigi Ricciato and Stefano Soatto , "***A Fast RANSAC–Based Registration Algorithm for Accurate Localization in Unknown Environments using LIDAR Measurements"***IEEE Conference on Automation Science and Engineering, Sept 22-25, 2007.

[3]Rahul Raguram, Jan-Michael Frahm, and Marc Pollefeys, "***A Comparative Analysis of RANSACTechniques Leading to Adaptive Real-Time Random Sample Consensus"***, The University of North Carolina, 2012.

[4] Mansoor Ahsan, Suhail Akhtar, Adnan Ali, Farrukh Mazhar, Muddssar Khalid***, "An algorithm for autonomous aerial navigation using MATLAB Mapping toolbox",***World Academy of Science, Technology and Engineering 66, pp. 858-861, 2012.

[5] Ryan Schaafsma***, "UAV imaging position errors",***Master Thesis, California State university, 2012.

[6] Serge Belongie, Jitendra Malik, Ian Puzicha , "***Shape matching and object recognition using shape contexts",*** IEEE, Transactions on pattern Analysis and Machine Intelligence, Vol. 24, 2002.

[7] V. Venkateshwar, Rama Chellappa, **"*Extraction of straight lines in aerial images*",** IEEE, Transactions on pattern Analysis and Machine Intelligence, Vol. 14, pp. 1111-1114, 1992.

[8] Mansoor Ahsan, Suhail Akhtar, Adnan Ali, Farrukh Mazhar, Muddssar Khalid, **"*An algorithm for autonomous aerial navigation using MATLAB Mapping toolbox*",** World Academy of Science, Technology and Engineering 66, pp. 858-861, 2012.

[9] Ryan Schaafsma, **"*UAV imaging position errors*",** Master Thesis, California State university, 2012.

[10]Stuart Ness, **"*Geo-registration of Remotely Sensed Imagery"*,** Department of Computer Science and Engineering, University of Minnesota.

[11] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, ***Estimation with Applications to Tracking and Navigation: Theory, Algorithms, and Software*,** Wiley, New York, July 2001.

[12] Dong-Gyu Sim, Rae-Hong Park, Rin-Chul Kim, Sang-Uk Lee, Ihn-Cheol Kim, "*Integrated Position Estimation Using Aerial Image sequences*", IEEE, Transactions on pattern Analysis and Machine Intelligence, Vol. 24, 2002.

[13]Seema.B.S, Hemanth Kumar, VPS Naidu, *"Application of Vision based Techniques for UAV Position Estimation"*, National Conference on Advance Researches in Electronics & Communication -2014, April 22-23, 2014.